Neurocomputing 101 (2013) 52–58

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semi-supervised spectral hashing for fast similarity search

Chengwei Yao, Jiajun Bu, Chenxia Wu, Gencai Chen *

College of Computer Science and Technology, Zhejiang University, No. 38, Zheda Road, Hangzhou, Zhejiang 310027, China

ARTICLE INFO

Article history:
Received 2 November 2011
Received in revised form
1 February 2012
Accepted 11 June 2012
Communicated by Y. Chang
Available online 29 August 2012

Keywords:
Hashing
Approximate nearest neighbor search
Dimensionality reduction
Embedding learning

ABSTRACT

Fast similarity search has been a key step in many large-scale computer vision and information retrieval tasks. Recently, there are a surge of research interests on the hashing-based techniques to allow approximate but highly efficient similarity search. Most existing hashing methods are unsupervised, which demonstrate the promising performance using the information of unlabeled data to generate binary codes. In this paper, we propose a novel semi-supervised hashing method to take into account the pairwise supervised information including must-link and cannot-link, and then maximize the information provided by each bit according to both the labeled data and the unlabeled data. Different from previous works on semi-supervised hashing, we use the square of the Euclidean distance to measure the Hamming distance, which leads to a more general Laplacian matrix based solution after the relaxation by removing the binary constraints. We also relax the orthogonality constraints to reduce the error when converting the real-value solution to the binary one. The experimental evaluations on three benchmark datasets show the superior performance of the proposed method over the state-of-the-art approaches.

1. Introduction

An explosive growth of data on the Internet brings both challenges and opportunities to traditional machine learning methods developed on small to median scale datasets due to the excessive cost in storage and processing. Among these methods fast nearest neighbor search (or similarity search) has been a key step in many large-scale computer vision and information retrieval tasks, such as near-duplicate detection [1], plagiarism analysis [2], collaborative filtering [3], caching [4] and content-based multimedia retrieval [5–8].

Early studies on the nearest neighbor search mainly focused on the tree-based techniques [9–13], which are typically quite memory-demanding, and the searching performance drops significantly on the data with high dimensionality. Recently, there are a surge of research interests on the hashing-based techniques [14–24], which have been presented to achieve the fast query time while substantially reduce the storage requirement.

Hashing aims to map semantically similar points in the database to similar codes (within a short Hamming distance). Such a compact binary coding technique is extremely fast to perform the similarity search. This is because (1) the encoded data is highly compressed thus can be loaded to the main memory and (2) the Hamming distance between two binary codes can be computed efficiently by using bitwise XOR operation and counting the number of set bits [25,26]. Computer can do this kind of computation extremely fast as we know.

In general, one can find the approximate nearest neighbors to a query in the Hamming space which have running times that are sublinear in the size of the database instead of a linear search by computing its similarity to all data points in the database. In such situations, one can simply return all the similar points that are hashed into a tight Hamming ball centered around the binary code of the query q. To further improve the retrieval effectiveness with few extra time, it is easy to re-rank a small set of returned "good" points based on their original features [27]. In addition, hashing-based similarity search can also be treated as the first stage of the classic non-parametric classification method like k-nearest-neighbor algorithm [28].

Most existing hashing methods are unsupervised which just employ the information of unlabeled data to extract binary codes. Locality-sensitive hashing (LSH) is one of the most popular methods [29,14,15]. Its extensions have also been developed in [16–18,30]. Besides the technique based on the random projection used in LSH, several other unsupervised methods are developed including semantic hashing [19,20]. However, in real applications, sometimes similarity (or distance) between data points is not defined with a simple metric. Metric similarity of data points may not preserve the semantic similarity.

It is possible for many applications to obtain a few data points with label information. Recently, several semi-supervised hashing

* Corresponding author.
E-mail addresses: yaochw@zju.edu.cn (C. Yao), bjj@zju.edu.cn (J. Bu), chenxiawu@zju.edu.cn, wutortoise@gmail.com (C. Wu), chengc@zju.edu.cn (G. Chen).

0925-2312/$ - see front matter © 2012 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.neucom.2012.06.035

methods [31,32] are proposed to take advantage of the information of both labeled and unlabeled data. In this paper, a novel semi-supervised spectral hashing method is proposed to take into account the pairwise supervised information including must-link and cannot-link, and then maximize the information provided by each bit according to both the labeled data and the unlabeled data. Different from previous works on semi-supervised hashing, we use the square of the Euclidean distance to measure the Hamming distance, which leads to a more general Laplacian matrix [33] based solution after the relaxation by removing the binary constraints. We also relax the orthogonality constraints to reduce the error when converting the real-value solution to the binary one motivated by Wang et al. [31]. The experimental evaluations on three benchmark datasets show the superior performance of the proposed method over the state-of-the-art approaches.

For better illustration, we give the following notations in this paper. A training dataset matrix is denoted as $X=[\mathbf{x}_1,\ldots,\mathbf{x}_n]\in R^{d\times n}$, each column of which represents a d-dimensional data point. Among the training data, the label information of a small fraction is known. According to the type of pairwise label relationship, each pair of labeled points is categorized into two groups: positive group (P), in which each pair of samples shares the same label; negative group (N), in which each pair of samples belongs to the different classes. Suppose that there are l points, $l<n$, each of which is associated with either P or N. Moreover, the matrix formed by these columns of X is denoted as $X_l\in R^{d\times l}$. The m-bit Hamming embeddings of X and X_l are denoted as $Y\in\{1,-1\}^{m\times n}$ and $Y_l\in\{1,-1\}^{m\times l}$ whose columns are the Hamming embeddings of corresponding points in X and X_l. E represents the expectation.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the related work on methods for fast similarity search. Section 3 presents our proposed Semi-Supervised Spectral Hashing (S3H) method. Section 4 provides several experimental validation on two benchmark datasets. Section 5 gives the conclusion and future work.

2. Related work

In this section, we first briefly review the fast neighbor search methods, and then study three state-of-the-art hashing methods: locality sensitive hashing, spectral hashing and semi-supervised hashing for the fast similarity search.

2.1. Fast nearest neighbor search

Methods for fast nearest neighbor search can roughly be divided into two categories.

One focused on finding the appropriate spatial partitions of the feature space via various tree structures including k–d trees [9], M-trees [11], cover trees [12], metric trees [13], k–d trees [30] and other related algorithms. These techniques attempt to speed up the nearest neighbor computation, but are typically quite memory-demanding. Moreover, the searching performance drops significantly on the data with high dimensionality. Additionally, the tree-based techniques can be degenerated into a linear scan in the worst case.

The other is the hashing-based method which is closely related to this paper. Locality-sensitive hashing [29,14,15] is one of the most popular methods. Also, there are some extensions for accommodating distances such as l_p norm [16], learned metric [17], and image kernels [18]. Several recent methods have explored to improve upon the random projection techniques used in LSH. Spectral Hashing (SH) is a promising unsupervised hashing method which has been shown to be very effective in encoding large-scale low-dimensional data since the important PCA directions are selected multiple times to create binary bits. Another popular unsupervised hashing method for fast document retrieval is Semantic Hashing [19]. To obtain the binary codes, Semantic Hashing binarizes the real-valued low-dimensional vectors obtained from dimensionality reduction techniques like Latent Semantic Indexing (LSI) [34,35] via thresholding which can also be called binarized-LSI.

2.2. Locality sensitive hashing

Informally, LSH [29] requires the randomized hash functions to guarantee that the collision probability of two vectors is inversely proportional to their distance. Such distance is defined according to the specific task. Since the similar points are assured to fall into the same hash bucket, one only need to retrieve those items in the database with which a novel query collides in the hash table. More precisely, the hash function $h(\cdot)$ from LSH family satisfies the following locality preserving property:

$$P(h(\mathbf{x}_i)=h(\mathbf{x}_j))=sim(\mathbf{x}_i,\mathbf{x}_j) \tag{1}$$

where $P(\cdot)$ represents the probability, sim denotes the similarity which can be directly linked to the distance function such as $sim(\mathbf{x}_i,\mathbf{x}_j)=\exp(-(\|\mathbf{x}_i-\mathbf{x}_j\|^2)/\sigma^2)$. A typical category of LSH functions consists of random projection and thresholds as

$$h(\mathbf{x})=\mathbf{sign}(\mathbf{w}^T\mathbf{x}-t) \tag{2}$$

where $\mathbf{w}$ is a random hyperplane and t is a random intercept. In [16], $\mathbf{w}$ is constructed by sampling each component of $\mathbf{w}$ randomly from a p-stable distribution, i.e., standard Gaussian. Though LSH guarantees the desirable collision probability, it is not very efficient in practice since it requires the multiple tables with long codes [29].

2.3. Spectral hashing

Due to the limitation of random projection-based LSH approach, machine learning techniques have been adapted to improve the efficacy of hashing. Particularly, SH [20] was recently proposed to obtain the compact binary codes for the approximate nearest neighbor search. SH seeks an m-bit Hamming embedding $Y\in\{1,-1\}^{m\times n}$ for the training points by minimizing

$$\begin{aligned}&\min_Y \frac{1}{2}\sum_{i,j=1}^{n}\|Y_i-Y_j\|^2S_{ij}=\mathbf{tr}\{YLY^\top\}\\ &\text{s.t.}\quad Y\in\{1,-1\}^{m\times n},\quad Y\mathbf{1}=0,\ YY^\top=nI_{m\times m}\end{aligned} \tag{3}$$

where Y_i is the ith column of Y representing the m-bit code for the corresponding data point, S is the $n\times n$ similarity matrix, and $D=\mathbf{diag}(S\mathbf{1})$ with $\mathbf{1}=[1,\ldots,1]^\top\in R^n$. The graph Laplacian is defined as $L=D-S$. The constraint $Y\mathbf{1}=0$ is imposed to maximize the information of each bit, which occurs when each bit leads to the balanced partitioning of the data. Another constraint $YY^\top=nI_{m\times m}$ forces m bits to be mutually uncorrelated in order to minimize redundancy among bits. SH assumes the data points in X had a uniform distribution. Then PCA transformation of the data points is calculated followed by the calculation of the top m smallest single eigenfunctions of the Laplacian, the sign of which yields the binary code.

2.4. Semi-supervised hashing

While being strikingly simple and efficient, SH suffers from a bold disadvantage—the assumption that the L_2 distance measures validly the semantic distance between data points, which essentially makes the approach unsupervised. For many real-world